Python: class ASV

asciidata.ASV = class ASV(UserList.UserList)

Methods defined here:

__getitem__(self, x)

```
Return row x as an instance of Row from this ASV instance
```

__init__(self)

__setitem__(self, x, row, field_names=None)

```
Set item x to row

row can be either a Row instance or a list. field_names is a list of
element in the row correnponds with and only makes sense if row is a
instance already has field names
```

append(self, row, field_names=None)

```
Append row to this ASV instance

row can be either a Row instance or a list. field_names is a list of
element in the row correnponds with and only makes sense if row is a
instance already has field names
```

extend(self, rows, field_names=None)

```
Append list rows to this ASV instance

rows must be a list. Elements in rows should either all be Row instan
if this is not the case the result of the extend is undefined.

row can be either a Row instance or a list. field_names is a list of
element in the row correnponds with and only makes sense if row is a
instance already has field names.
```

get_field_names(self)

```
Return this ASV instances field names

Returns None if this ASV instance does not have any field names
```

input(self, data, input_class, *args, **kwargs)

```
Process input data using input_class

Although the input_class can specify what type data should be, in gen
string.

See 'input and output classes' in the main documentation for more dis
input_class should refer to.
```

input_from_file(self, input_file, input_class, *args, **kwargs)

Process input data from a named file

This is a convenience method. input_file should be the name of a read
input method for details of the other arguments

output(self, output_class, *args, **kwargs)
Create output data using output_class

output_to_file(self, output_file, output_class, *args, **kwargs)
Output data straight to a named file

This is a convenience method for the output method

set_field_names(self, field_names)
Set the field names for this ASV instance

field_names must be a list of strings.

You can not set field names if you they have already been set (either
this method or indirectly by another method such as input) or if this
holds data.

Methods inherited from UserList.UserList:

__add__(self, other)

__cmp__(self, other)

__contains__(self, item)

__delitem__(self, i)

__delslice__(self, i, j)

__eq__(self, other)

__ge__(self, other)

__getslice__(self, i, j)

__gt__(self, other)

__iadd__(self, other)

__imul__(self, n)

__le__(self, other)

__len__(self)

__lt__(self, other)

__mul__(self, n)

__ne__(self, other)

__radd__(self, other)

__repr__(self)

__rmul__ = __mul__(self, n)

__setslice__(self, i, j, other)

count(self, item)

index(self, item, *args)

insert(self, i, item)

pop(self, i=−1)

remove(self, item)

reverse(self)

sort(self, *args, **kwds)